AFRL-IF-RS-TR-2002-155
Final Technical Report
July 2002

UNDERSTANDING AND EXPLOITING HIERARCHY

University of Oregon

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. F081

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2002-155 has been reviewed and is approved for publication.

APPROVED: John F. Lemmer

JOHN F. LEMMER
Project Engineer

FOR THE DIRECTOR: Michael L. Talbert

MICHAEL L. TALBERT, Technical Advisor
Information Technology Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	JULY 2002	Final Jun 97 – Jun 00

4. TITLE AND SUBTITLE
UNDERSTANDING AND EXPLOITING HIERARCHY

5. FUNDING NUMBERS
C - F30602-97-1-0294
PE - 62301E/63670F
PR - COMP
TA - T3
WU - 02

6. AUTHOR(S)
David W. Etherington, Matthew L. Ginsberg, and Brian Drabble

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Oregon
CIRL, 1269 University of Oregon
Eugene Oregon 97403-1269

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington Virginia 22203-1714

AFRL/IFTB
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2002-155

11. SUPPLEMENTARY NOTES
AFRL Project Engineer: John F. Lemmer/IFTB/(315) 330-3655/ John.Lemmer@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT	12b. DISTRIBUTION CODE
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.	

13. ABSTRACT ***(Maximum 200 Words)***

The project made important progress in the three key areas it set out to address: information content, information flow and information processing, showing that explicit modeling of information, its content and the ways it changes, can provide a powerful means of handling large distributed problems.

Among these, the use of generic task description templates greatly improves the agent-tasking process by making explicit the constraints and dependencies between tasks. Such task models allow algorithms to understand potential tradeoffs and identify ways tasks can be modified to suit the changing environment.

The technologies and ideas developed during the project have been successfully applied to problems in mission planning and ISR management. In particular, the DEOS system developed under the project offers faster, more flexible solutions than those available using current technologies.

Research on the information-processing aspects of process management highlighted several new approaches, particularly exploiting phase transitions. These are naturally occurring "computational cliffs" in problems that represent the point where problems transition from being manageable to be being very difficult to solve. Many important problems fall in this transition region, making the potential payoff of this work very high.

14. SUBJECT TERMS
Resource Management, Task Models, Agent Tasking, Search, Robustness, Constraint Satisfaction, Hierarchy Modeling

15. NUMBER OF PAGES
47

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

Understanding and Exploiting Hierarchy

Brian Drabble
David W. Etherington
Matthew L. Ginsberg

CIRL
1269 University of Oregon
Eugene, OR 97403-1269
drabble@cirl.uoregon.edu

January 26, 2001

Final Report for contract number: F30602-97-1-0294

Period of performance: 6/6/97 to 6/5/00

Point of contact: Brian Drabble, Director
CIRL
1269 University of Oregon
Eugene, OR 97403-1269
fax: 541-346-0474
phone: 541-346-0473
email: drabble@cirl.uoregon.edu

Contents

i Cover Page **1**

ii List of Figures **3**

1 Introduction **4**

1.1 Project Overview 4

1.2 Summary of Accomplishments 5

1.3 External Interactions 8

1.4 Report Overview 8

2 Information Content **9**

2.1 Activity Description Model 9

2.2 Process Objects and Products Model 10

2.3 Information Content Model 11

2.4 Agent Capability Modeling 12

3 Information Flows **14**

3.1 Mission Planning Task Model 14

3.2 Campaign Management Task Model 16

4 Information Processing **19**

5 Empirical Studies **25**

5.1 The EBO Domain 26

5.2 The AIM Domain 27

6 External Interactions **27**

7 Summary and Future Work **29**

8 Published Papers **34**

List of Figures

1 PRFER Task Model Example 16
2 PAER Task Model Example 18

1 Introduction

This final report is provided to Rome Laboratory and the Defense Advanced Research Projects Agency (DARPA) for the contract F30602-97-1-0294, entitled “Understanding and Exploiting Hierarchy”. The project was scheduled for the three year period from June 6th, 1997 through June 5th,t2000.

At the request of the program manager, resources were diverted from the project to investigate a number of aircraft-routing problems of interest to AFRL. This report describes work completed under the original proposal but does not cover the work in aircraft routing, which is described elsewhere [29].

1.1 Project Overview

Modern military operations call for planning to be done by a hierarchy of coordinated planning teams that deal with different aspects of the problem, possibly at different levels of abstraction, and interact on an asynchronous, information-driven basis. By contrast, research in planning has generally treated planning as a monolithic process. The traditional focus of research in planning and scheduling in both artificial intelligence and operations research has been on solving one-shot optimization problems: for a given input, one is expected to find values for a set of variables that collectively maximize some objective function. This approach has led to the development of systems that solve isolated problem instances well, but are fundamentally ill-suited for deployment within hierarchical networks of collaborative planning teams. Modern military operations thus require the development of new formal and algorithmic approaches to planning and scheduling that are inherently dynamic and modular.

We know that effective hierarchies are possible, since they are ubiquitous in our society. Generals and colonels interoperate well because their responsibilities are structured according to a natural and effective, mutually understood, military hierarchy that has been developed through centuries of refinement. Hierarchical software components interoperate far less effectively because there is no such understanding to provide overarching design principles to constrain their construction.

The fundamental goal of this project was to understand the sort of information driven hierarchy of interoperating planning modules military planning needs, to formally characterize it, and thereby exploit it. Rather than take existing technology and try to patch it to the point where it can support these goals, we started from first principles by studying the structure of hierarchy itself and determining what that structure says about how planning systems should be built. We used the resulting understanding of hierarchy to develop guidelines such that individual software components conforming to these guidelines can be expected to interoperate effectively.

By exploiting hierarchy, we make it possible for problems to be solved by dividing them into manageable pieces, with each piece being solved by a dedicated software module or person. This sort of approach is essential in the dynamic, collaborative environment that will be faced by the military forces in the future.

A first-principles approach to understanding the military notion of hierarchy immediately produces the insight that communication must provide the fundamental organizing principle around which everything else is based. What defines the hierarchy is who talks to whom; once this structure is clear, other issues can begin to be understood. Furthermore, the information-driven nature of the process dictates that the communication model must be asynchronous, since information can arrive at any time, from superiors, subordinates, or peers. All of the entities in the hierarchy are operating in parallel, and the communication model must support this.

The understanding of hierarchy we have developed separates the mappings that communicate information between entities, the information that is communicated, and how the information is processed within entities, and opens each up to analysis. Our approach begins by defining the information pathways within the hierarchy, separating the communication between superiors and subordinates (or between peers) from the information processing done within entities. This model can accommodate both the asynchronous nature of the process, and the fact that the process is not a strict hierarchy (much less a linear hierarchy, as some AI planning work assumes).

By formalizing an asynchronous communication model between agents, our framework naturally supports just-in-time tasking, horizon-dependent plan granularity, and rolling planning. It also makes it possible to take advantage of the parallelism implicit in a hierarchical structure like that envisioned in military planning, and allows work to proceed on plans at multiple levels at the same time. Moreover, because communication between peers is a basic part of the framework, it supports the use of coordinated planning teams. The information-driven nature of the military structure means that this characterization of the flow of information in a hierarchy, and the separation of information flow from the information processing done within an entity, is foundational work.

1.2 Summary of Accomplishments

The main focus of the project was on the development of ideas in workflow and task management. The effort was centered around three key areas: information modeling, information flows, and information processing. The accomplishments of the project in each of the three key areas are summarized below and discussed in detail in later sections of the report.

Information Modeling

We developed information-modeling techniques that allow the pertinent attributes of various process products (e.g., documents, orders, messages, guidance) to be captured and used to trigger and monitor activities in the hierarchy. In addition, the techniques allow modeling of the dynamic modification of information as it progresses from one group to another. For example, the JIPT meeting[1] can start with a recommended JITPL and turn it into an approved JIPTL. By understanding how information is created, modified, updated and authorized (referred to as the CRUA model) it becomes possible to build better information-routing processes. For example, knowing the weaponeering group can begin their planning with a recommended JIPTL but cannot finish until it has been checked against the approved JIPTL allows a dramatic improvement over the simple stove-piped military processes.

The information model has been extensively tested in the ISR and mission management domains. It was used as part of a major demonstration in DARPA's AIM program in the second quarter of 1999. The model demonstrated how intelligent workflow management could be used to develop, monitor, and coordinate the activities in the sensor-planning and information-processing steps. The model also proved to be very useful in the mission-management domain in providing a link between the schedule-generation and execution processes. For example, the coordination of different mission types (e.g., tankering, ISR, CAP, airlift, SEAD) is controlled by the exchange of information messages. By modeling this flow, it is possible to understand how execution time failures (e.g., an AWACS aborting on take off) should be communicated and who should be involved.

Information Flow

We developed task models that capture the ways tasks are assigned and the dependencies different tasks impose on one another. Task models capture meta-information about tasks and reflect the regular structure of tasks in a domain. They have been successfully applied to both the ISR and mission-management domains and have allowed users to gain a clearer understanding how tasks interact with, and/or impose dependencies on, other tasks. For example, a strike mission may have a higher probability of success if a SEAD mission is sent to accompany it. This raises the need for a SEAD mission to be planned, which is the function of a different group in the air campaign planning hierarchy. If the SEAD planners cannot provide such a mission, this should be communicated back to those missions that requested SEAD and not to the entire planning hierarchy. Task models have also been applied to a problem

[1] The Joint Integrated Prioritized Targeting meeting develops the JIPT List for the the air tasking order.

from the banking industry, modeling information intensive tasks such as mortgage applications. Coupled with the information model, task models provide a flexible and extensible way of describing workflow problems while at the same time allowing the reasoning engine to identify the different trade-offs and options between different resource assignments.

Information Processing

We developed the DEOS engine as a test-bed for a number of task-management and agent-coordination techniques. DEOS was extensively tested in the ISR (intelligence, surveillance and reconnaissance) and mission-management domains and was an integral component to AFRL's Effects Based Operations (EBO) jump-start demonstration. DEOS explored several new scheduling and resource-allocation strategies that can handle highly dynamic and reactive domains. The information processing research identified key factors in deciding ways in which tasks should be decomposed and the formation of coalitions of agents to deal with them. The

A correct formulation of tasking ensures that "computational cliffs" are avoided. Ideas were explored to highlight the importance of building solution clusters, which allow tasked agents flexibility in dealing with their assigned tasks. This flexibility allowed for more robust solutions that had a higher expectation of actually working, and at the same time reduced the communications bandwidth. These ideas are now being explored further through other efforts funded by DARPA's ANTs and CoABS projects.

Other Activities

In addition to the achievements outlined above, a significant fraction of our efforts under this award were redirected, at the request of program management, to developing a practical application of search technology to a real military problem: aircraft routing for the USAF's Air Mobility Command (AMC) [29]. We developed an extensive proof-of-concept system that demonstrated that search technology could produce dramatic (estimated 1-3% of 750M gal) fuel savings for AMC. We then supported the transfer of this technology to On Time Systems, Inc., which is preparing to deliver a production route-planner based on this technology to AMC. Details of this work, which amounts to a successful transfer of ARPI technology to operations, are not included in this report as they do not directly bear on the research goals of the award.

1.3 External Interactions

A substantial effort was devoted to interacting with other groups and DARPA projects to ensure we examined as many different hierarchies and domains as possible. We developed relationships with DARPA's Advanced ISR Management (AIM) and Control of Agent Based Systems (CoABS) projects and with AFRL's Effects Based Operations (EBO) jump-start demonstration.

- AIM The AIM program provided several large examples of ISR hierarchies that gave valuable insights into how ISR missions are planned and their results deseminated. Several components of this work were evaluated through the AIM program.

- CoABS The CoABS program's mixed-initiative agent planning experiment (MIATA TIE) provided a number of agent tasking problems that demonstrate agents working collaboratively to solve a hurricane relief scenario. This provided several insights on how hierarchies are adapted and modified to meet the needs of an operation. Several ideas and concepts are currently being transfered and evaluated in this program.

- EBO The EBO program provided the bulk of the hierarchies we studied as well as the main test-bed for our ideas. DEOS was incorporated into the jump-start program and was demonstrated at Langley AFB in October 2000.

Contacts have been made with several companies, including Logicon, Lockheed Martin and Boeing, concerning the potential use DEOS in mission management systems. Lockheed Martin has selected DEOS as its preferred technology for mission planning for the CVN-77 ("Nimitz after next" carrier project) which it won in February 2000. Additionally, DEOS is currently being considered for the bid headed by Bath Iron Works for the DD-21 land-bombardment destroyer project. DEOS is being considered for the mission management of unpiloted weapons (e.g., cruise missiles, shells). Boeing's Military Aircraft division is developing several EBO applications for the Navy and is currently evaluating DEOS. Full details of these interactions can be found in Section 6.

1.4 Report Overview

The report is structured around the three main topics investigated during the project. Section 2 describes the modeling of information within hierarchies. Section 3 describes how information is routed and directed to different tasks. Section 4 describes the workflow engine that was developed to use the information flow model, and Section 5 describes a number of experiments conducted in ISR and mission-management. Section 6 describes interactions with other DARPA projects and external groups and companies, and Section 7 provides a summary of the project, its achievements, and future work.

2 Information Content

This section provides an overview of the information-content models that were developed to support the task-description and decomposition problem.

2.1 Activity Description Model

As described earlier, one of the main aims of the project was to identify a general framework within which tasks could be described, scheduled and tracked. The project succeeded in this aim and developed a task-description model that can describe tasks at either the process level (e.g., the process of putting together an ATO) or the domain level (e.g., which targets need to be struck and when). Each of the tasks at either level is specified using "task verbs" indicating the task to be performed. The tasks are described in the following form, referred to as the "Verb/Noun/Qualifiers model" (VNQ):

- **Verb**: the task to be carried out (e.g., analyze, develop, attrit)
- **Noun Phrase(s)**: one or more noun phrases describing the object(s) or products on which the activity is being performed (e.g., prioritized target list)
- **Qualifier(s)**: zero, one, or more qualifiers constraining how the activity is performed (e.g., time/resource limits)

The process products are modeled as resources that are created, modified, used, and authorized within the process. This is referred to as the CRUA model and provides the basic framework for task triggering. Examples of process products are documents, reports, orders, letters, and communications (formal or informal). Authority relationships and other conditions are also modeled and can be used as an extension of the basic mechanism.

The VNQ model and process products have been encoded using the ACT representation developed by SRI [28], and can be directly executed by the Procedural Reasoning System (PRS) [30, 17]. The model is hierarchical and provides a rich scheme for both the representation of normative processes and the derivation of new processes based on AI reasoning and planning. The same VNQ model can also be used to describe the capabilities of the agents in the domain. Details of the agent capability descriptions are provided later in this report. The VNQ model was originally developed for the ACP domain [10]. Development was undertaken in collaboration with DARPA's AIM program and details can be found elsewhere [3]. This framework is general enough to be applicable across various military domains as in manufacturing, logistics and supply-chain management. Details of the verbs and process products are given in the following sections.

2.2 Process Objects and Products Model

The process objects and products model is developed through a series of matrices that define, for each task in the process, the process products that are created, read, updated and approved (referred to as the CRUA matrices). The matrices also allow for the identification of the process products that could be associated with a given verb and/or qualifier. For example, the verb "review" can be applied to the JFACC Guidance Letter but the verb "critique" could not. This analysis also identifies potential classes of process products and associated values that could provide additional structure to the task models. For example, all process products that refer to a list of targets (e.g., candidate target list, target nominations list, service target nominations, JIPTL and JIPTL cut-off) could be grouped together with a verb (e.g., all these lists can be analyzed but not critiqued). The CRUA matrices were also used to identify the other constraints that are involved in each step in a process. The main constraints modeled are:

- **resource**: the agents (human and/or software) required to carry out the activity.
- **temporal**: either qualitative (e.g., "the start of activity A precedes the start of activity B") or quantitative (e.g., "activity A must end no later than 16 hours after the start of conflict").
- **authorities**: specific authorities that must be obtained before an activity can start or finish (e.g., presidential authority must be given before the operation can begin).

An example of part of the task-to-process-product association is given in Table 1. However, some tasks refer to objects in the domain without associating them with a specific process product. For example, steps such as "group targets" and "deconflict

Verb	Noun Phrase(s)	Qualifier Phrase
Deconflict	ACM Requests	
Finalize	Air Control Order Special Instructions Air Tasking Order	 Quality Control
Produce	Air Tasking Order Target Groupings CAS Sortie Allocations Potential Target List Initial Target Nomination List Weaponeering Assessment Weaponeering Force Assessment Mission Support Requirements	 Broad
Release	Air Tasking Order	
Consider	Target and Route Threats	

Table 1: Part of the Verb/Noun(s)/Qualifier(s) Table

airspace" refer to objects in the domain rather than process products. Features of the ACP domain such as airspace, targets and ground features exist in the real world and are not created by the ACP process. To be able to handle these features a class of process objects was added to the model. The distinction between process objects and process products is that process objects are not created by a process but can be used, modified and consumed in the same way as process products.

2.3 Information Content Model

Using the process product features identified from the CRUA matrices, it is possible to group features into classes and associate with each class a descriptor type. For example, the features "available" and "not available" could be grouped together to form a single feature "availability" that can take one of these values. These features are used to form an ontology of primitive process product features that could is in turn used as the building blocks for more complex reasoning about the status of process products. For example, the status of the document could be "available, compound, published, draft". Features such as status are not part of the primitive ontology of process products but are instead composed of primitive features. In addition to identifying the features, it was also necessary to identify the potential values the attribute can take (e.g., the attribute `availability` can take the values `available` or `unavailable`). An example set of features and values is provided below.

- **Process Product Type**: the type of product, could take values such as ATO, ACO, JIPTL.
- **Availability**: the availability of the process product, simply defines whether the process product exists. It could take values such as `unavailable` or `available`.
- **Review-Status**: the review status of the process product as it is reviewed and passed through the process. It could take values such as `on-going`,`cut-off`, `final`.

The agents in any large scale problem will be working not only on distinct tasks but also at different levels of abstraction and with varying temporal horizons. This complicates the problem of information sharing and communication between tasks. In one direction, strategic objectives have to be prioritized and transformed into feasible tasks (e.g., prioritize current target list) and then into specific domain level tasks (e.g., attack this bridge on D+4). In the other direction, information about the satisfaction of those tasks must be abstracted up to the strategic level and related to the initial or current objectives. For example, how does information from a single asset-management agent about the failure to schedule a single strike mission feed back to the agent concerned with targeting development? The single asset manager is concerned with short term, detailed information, the strategy manager is concerned with longer term achievement of intelligence objectives.

2.4 Agent Capability Modeling

The agents in any process management problem will be a mixture of humans and software systems. Each will have one or more capabilities that can be used by the workflow manager to carry out steps in the process. The function of the process manager is to identify the most appropriate agent amongst those with the required capabilities. To do this, the workflow manager needs to make extensive use of an agent capabilities model. This model describes the capabilities of both single agents and clusters of agents.

Clusters of agents may be defined as static bodies (e.g., in the ACP domain, agents from the different branches meet daily in the Joint Target Coordination Board (JTCB)). Other clusters would be more dynamic and exist for a specific purpose: once that purpose was finished the cluster would be disbanded (e.g., mission-specific operations). The agent capabilities model is based on the same verb/noun phrase(s)/qualifier phrase(s) (VNQ) model used to capture the steps in the ISR process. This model was successfully used in the ACP domain to capture the capabilities of systems such as TRAINS [16], SIPE [37], O-Plan [35], INSPECT [33] and OPIS [32]. Some simple experiments were conducted using the model to capture the capabilities of human agents (i.e., air campaign planners at CHECKMATE). These experiments successfully showed that the relevant capabilities could be captured using the VNQ model, and that the model provided an intuitive way of viewing those capabilities.

The model will likely need to be augmented to deal with the dynamic nature of agent capabilities, which can be explicitly gained or lost over time. In addition it is expected that the model will need to take into account other factors such as length of time since a human agent last carried out the task and time taken to retrain. Using such a capabilities model makes it possible to:

- identify the appropriate level of workflow description needed to correctly task the agent, and
- define a capabilities model that would be useful across a number of different military programs.

For example, in the ACP domain, an agent with the capability to develop an air tasking order does not need the task broken down to the lowest level of detail and the agents allocated in advance. However, if no such agent is available, the workflow manager should break the task down and identify the appropriate agents. This level of functionality and targeting of the task specification is not available in current workflow systems.

Domain-level descriptions will differ across different problems. In the ACP domain the descriptions include destroy, attrit and paralyze, [36, 34] while in the logistics domain they would include deliver, store, distribute. However, the process level stays the same. Both domains contain descriptions such as assign, prioritize, publish. The verb portion of the VNQ model has already been used by Dave Hess of SAIC to describe the top level functions of the Air Force's C2 functions [24]. While the investigations carried out by Hess were only at the top level of the C2 functions, he was confident that the model could be applied at the lower levels of the C2 hierarchy (personal communication).

For each agent in the system, a separate capability description is defined. This provides the process manager with a more detailed model of the capabilities of agents than was previously possible. The components of the template include the agent,s role, who is allowed to task it, temporal and resource constraints it imposes, the process products it creates, reads, updates or authorizes, and the trade-offs between its performance and the processing time it is allocated [3].

3 Information Flows

This section describes the information flow models that were developed and evaluated during this project. A key aspect of the information flow issue is making sure the tasks allocated to groups/agents are understandable and that the receiving agent is aware of the constraints associated with a task. One approach could be to add all constraints to a task when it is allocated to an agent (e.g., send a copy of the ATO with each mission). Obviously, this is impractical as it would overload the receiving agent and overwhelm the available bandwidth. One advantage of tasks in military hierarchies is that they tend to have a regular structure depending on the problem being addressed and the groups involved. For example, AAR tanker planners do not tend to receive tasks concerning building bridges. Instead, they receive requests for gas at certain times and locations. The regular structure of these tasks allows a scheduling algorithm to identify the pertinent constraints associated with a task, and more importantly, between tasks. For example, it is pointless to schedule an AAR rendez vous for a mission that has been canceled. In addition, it is pointless having the AAR planners spend several hours planning AAR orbits when the planes need gas in the next sixty minutes.

In response to these requirements, we developed the concept of flexible task models that capture the regular structure of the tasks in a domain and allow constraints on and between them to be dynamically posted and updated. The definition of a task model can contain both looping and condition structures (e.g., repeat until a process product becomes available, repeat for ten iterations). The tasking agent can use the task models to place restrictions on the entire task (e.g., the answer should be produced in less than 5 minutes) or on components of the task model (e.g., do not take longer than 2 minutes to file the results). To date we have developed two different task models.

3.1 Mission Planning Task Model

The hierarchies provided by the EBO and AIM programs demonstrated that the tasks in a mission-planning process could be divided into five subtasks or blocks and each of the blocks was common to any mission being flown (e.g., AAR, AWACS, strike missions, airlift missions). The task model developed for mission planning is referred to as the PRFER task model:

- **P**lan: Time taken for the pilot to plan the mission. Once a plan has been identified it is inserted in the slot for other workflow tasks to examine and check.

- **R**eady: Time taken to prepare the plane for the mission.
- **F**ly: Time taken to get to the mission objective[2].
- **E**xecute: Time taken to execute the mission (e.g., drop weapons, unload food pallets).
- **R**econstitute: Time taken to return to base (or friendly territory) and turn the aircraft around once for its next mission.

Each task is associated with a task specification block (TSB) which is allowed to "breathe" as changes in the domain are reflected as changes in one or more of the TSB's subblocks. For example, if the aircraft chosen for the mission develops a failure during its ready time, the "ready" subtask will expand to accommodate the extra time. To handle this change, the scheduling engine may decide to substitute for the aircraft if a spare aircraft exists or another can be re-weaponed in time. If no other aircraft is available, then the scheduler may try to reduce the time of the "execute" block to accommodate the longer-than-expected "ready" time. For example, if the aircraft is tasked with a food drop and the current method is to land and off-load the supplies then the "Execute" block would be set to three hours. If the mission were changed to an air drop, (i.e., drop the pallets via parachutes) then the "Execute" block would drop to 30 minutes but the "Ready" time would increase due to the time take to change the food pallets to an air-drop configuration. Breaking the task into subcomponents allows a planner to focus on ways to improve the schedule and recover from changes occurring in the domain and task. New TSBs can be added to the schedule as needed and removed just as easily should the decision be reversed [12]. The DEOS scheduler described later keeps track of the criteria under which the task was spawned. On each cycle, the scheduler quickly reexamines the need and removes the task where appropriate. For example, if the food drop is being carried out using C-141 Starlifter aircraft then the food pallets must be loaded using a K-1 lifter. However, if C-5 Galaxy aircraft are used, then a K-1 is not needed. Once an aircraft is chosen, the scheduler can examine the "Ready" block and examine needs for the aircraft type. Other needs, such as fuel, can be attached to other subtasks (e.g., "Fly").

By representing the schedule as a series of TSBs, it becomes possible to create different perspectives and avoid the problem of resources being informed of changes that have no effect on them. Figure 1 shows a number of tasks being coordinated using the PRFER model.

If the AWACS aircraft that is supplying air traffic control for the C-141s, the fighter protection and the air-to-air refueling tankers aborts on take-off, the immediate effect

[2] This can be replaced by a "drive" or "sail" block for operations using land or sea transport.

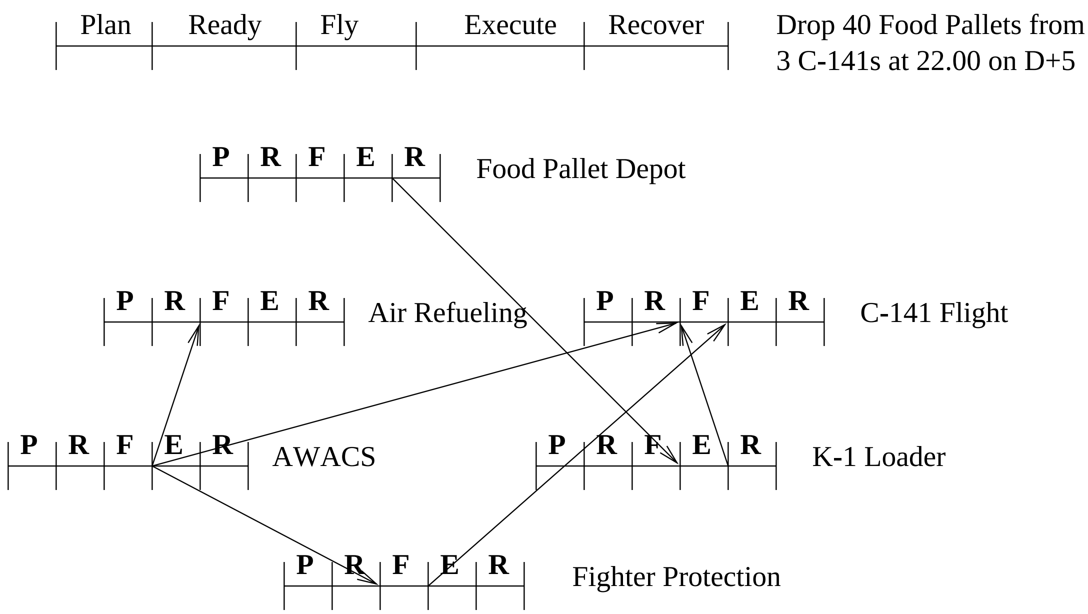

Figure 1: PRFER Task Model Example

is that its "Fly" block expands thus pushing its "Execute" block out further. This has the knock-on effect of pushing the "Fly" blocks of the tankers, C-141s and fighters out as well. If the C-141s are on the ground then, provided no later missions are impacted, they can remain there. If the C-141s are in the air they can loiter providing they have enough fuel; if not, a new task should be spawned to provide them with enough fuel. The operator in the food depot is informed that he has longer to provide the food pallets and is hidden from why his "Execute" block has expanded. Should it be required, it is possible to follow the chain of reasoning back from the K-1 task to the AWACS problem. Normally, the K-1 operator can extract his own tasks and not be bothered by the other tasks in the schedule. In the case of the K-1 loader the PRFER model can still be applied except that "Fly" is replaced by "Drive" and the "Recover" step consists of refueling, cleaning and resetting.

3.2 Campaign Management Task Model

The hierarchies provided by the AIM program demonstrated that the tasks in a domain dominated by information management could be divided into four sub-tasks or blocks and each of the blocks was common to any task being assigned (e.g., collection missions, processing requirements, exploitation and dissemination needs). The same generic task breakdown was also observed in the ATO generation process in which the objectives and goals of the commander are translated into target systems, sets and eventually individual targets. The task model developed for the campaign management process is referred to as the PAER task model:

- **P**lan: Time to plan the task. Once a plan has been identified, it is inserted in the slot for other tasks to examine and check.
- **A**cquire: Time to acquire the information (e.g., process products[3] necessary to carry out the task). This also specifies the resources (e.g., platforms, software packages) needed to run the task.
- **E**xecute: Time to carry out the alloted task.
- **R**eport: Time to file or report the results of the task.

Again, each task is associated with a TSB and can be handled with the same scheduling algorithms as the PRFER model. During the planning sub-task a number of requirements are identified and posted to the "Acquire" block. The workflow engine can generate new TSBs for these if there are no other tasks in the current schedule providing them. Alternatively, if another TSB is expected to generate the required document then its "Report" block can be modified by the workflow engine to provide an addition copy. In this way the execution of the PAER block is a partial order with some information gathering being carried out before all planning is complete. In addition, by identifying the information passing between process-level TSBs and domain-level TSBs it becomes possible to route the right information through the hierarchy. Figure 2 shows an example of how a PRFER task launched from an information product generated in a PAER task can be traced back should a problem occur with the PRFER task.

The PAER model provides an explicit model of delegation and authority through the planning information stored in the "Plan" TSB. The example shown in Figure 2 shows the agent delegating four separate tasks to its subordinate agents (the small activity plan shown in the "Plan" sub block). Through the delegation mechanism, the subordinate agents know whom to inform should a problem occur (i.e., a late-running task, unavailability of a report). Information passing through the different TSBs is used to coordinate and trigger other TSBs. For example, once a target list moves from recommended to approved (through a vetting task) then other TSBs are triggered. However, the workflow engine could identify that a task can execute with the recommended target list to get a "jump-start" and can finish its processing once a check has been made against the approved target list (in case changes have been made). This allows the workflow engine to trade off accuracy for time (i.e., the

[3]Process products are the orders, documents, reports, letter, etc, that are used to coordinate the workflow process.

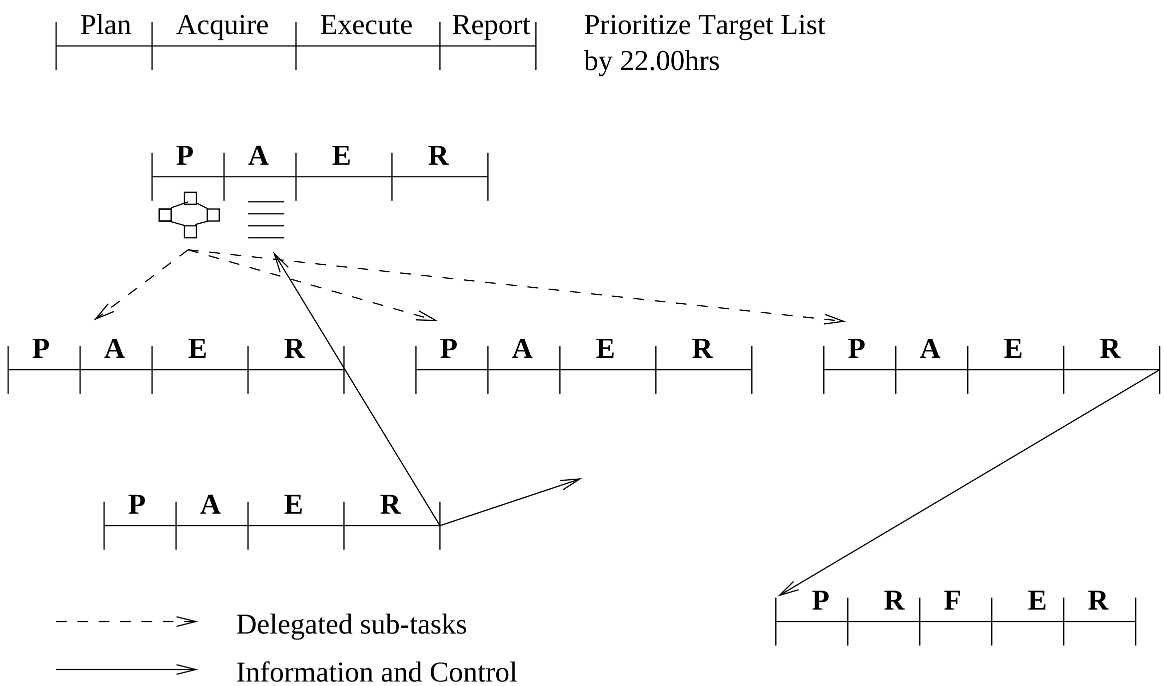

Figure 2: PAER Task Model Example

recommended target list may change but, if not, the process can be considerably shortened). It is this type of adaptive workflow that the PAER and PRFER models are designed to support. The following list describes several situations in which the PAER and PRFER tasks models can be used to handle external and internal events. The examples are drawn from experiments with the AIM ISR domain.

- **Change in a major process product (e.g., commander's guidance)**:
  This triggers an event in the workflow manager noting that a primary process product has changed status unexpectedly and that action should be taken. This means identifying the TSBs in the process that have the commander's guidance in their "Acquire" block. This would result in some tasks being suspended and others re-tasked to make use of the new information. If a secondary process product supporting the development of a new commander's guidance is delayed, then the "Acquire" block of any task using it would be extended by the time required. This will have potential knock-on effects on other tasks in the process. Should the support document be delayed indefinitely, an alternative or inferior (e.g. the previous day's) version would be used instead.

- **Initial loss of a domain asset (e.g., U2 aircraft)**:
  As with the previous example, the assumption is that the change requires one

or more current/planned tasks to be modified. This is achieved by identifying the tasks impacted by the lost asset. The workflow engine identifies the process products associated with the lost asset. These are then updated and changed as appropriate (see above). The loss of the asset may also have an impact on the current processing of the system. For example, if an asset scheduler was carrying out a full schedule with data including the lost asset, then an option might be to reduce the scheduler's task to a feasibility estimate to try and determine the consequence of the lost asset. This means reducing the "execute" block to reflect a feasibility probe rather than the full schedule.

- **Loss of a process-level asset (e.g., loss of asset scheduler)**:
  This involves adding new tasks and in some cases decomposing tasks to lower levels should there be no equivalent asset available. This can be handled simply by putting a new plan in the "Plan" slot and modifying its process product needs. For example, if some process products have been created or updated, the new task can ignore this need. If, however, the task is yet to start then a simple substitution may be adequate.

- **Upgrade in the forces in the conflict and the need for feasibility estimates of the ISR needs**:
  The above examples show the workflow engine repairing events occurring in an already-assigned series of tasks. There will be situations in which new tasks and requirements are added to the system. Such a situation would arise if a feasibility estimate is needed for future ISR requirements while trying to maintain the overall picture of the ISR process. This would mean identifying ways of scaling back current effort through changing "Plan" and "Execute" blocks of required TSBs.

4 Information Processing

During the project, a number of different resource-allocation strategies were explored. The aim was to identify algorithms that are suited to different types of allocation problems that arise in process management. Problem attributes included precedence constraints, probabilistic outcomes and time-limited reasoning. To support this aim we undertook a mix of theoretical and empirical studies that attempted to identify the kernel of the computational problem and to then evaluate algorithms through application to one or more of our experimental domains. The areas investigated include:

- **Squeaky Wheel Optimization**:
  Squeaky Wheel Optimization (SWO) allows many schedules to be generated and

analyzed very quickly. This allows SWO to quickly adapt to changes occurring in either the mission and/or the environment. This provides process-management technologies with the ability to keep agent taskings up-to-date and to modify the taskings as needs demand. There is no point generating taskings for out-of-date situations and producing results that are pointless or inaccurate. The aim of our work was to evaluate SWO on process management problems in which there were many dynamic changes and to develop ways in which the SWO algorithm could handle these changes while still maintaining its ability to quickly generate meaningful schedules.

- **Schedule Packing**:
  In many process-management applications, two of the key needs are to ensure that tasks are accomplished as quickly as possible (i.e., minimize the makespan of the process) and that schedules are developed that allow processes to be repeated in a cyclic manner. Schedule packing was developed as a makespan minimizer and tested in several manufacturing domains. The aim of our work was to investigate its application to process management, in particular its ability to develop high-quality cyclic solutions to large complex information management problems.

- **Limited Discrepancy Search**:
  The ability to generate qualitatively different solutions is a key need in process management. There is little point in applying similar solutions to a failed situation. The LDS search algorithm is able to search in many more parts of the search space than other search algorithms, while at the same time generating high-quality solutions. The aim of our work was to investigate whether LDS could handle process management problems in which qualitatively different solutions were needed.

- **Tractable reasoning**:
  Many problems faced by process management systems are extremely time limited (i.e., a solution is needed very quickly). The aim of our work was to characterize such problems and provide a semantics that could be used to develop "anytime" algorithms. Such a characterization could have large benefits for process management systems. For example, problems could be run for different durations and the trade-off between solution time and solution quality analyzed (i.e., longer time should ensure better quality solutions).

- **Phase transitions**:
  One key problem in process management is how to divide problems in a way that provides for computational efficiency There is no point in dividing a problem so that all the allocated time given to an agent is spent negotiating over

different decisions). Our work was to investigated ways in which computational cliffs could be identified and if possible avoided. Again, this would have large potential benefits for process-management systems as it would provide users and systems with invaluable advice on how problems should be decomposed and tasks allocated.

Some of these ideas were incorporated into demonstrations while others remain for further study, but they did provide valuable insight into the ways in which systems such as DEOS could be improved. In developing this mix of studies, our overall theme of how agents are tasked and how the problems should be divided up remained the central focus. Details of the different studies are provided below.

Squeaky Wheel Optimization

The original Squeaky Wheel Optimization (SWO) algorithm [26] was extensively modified in two ways:

Temporal Robustness

A better model of time and temporal reasoning was added to allow the scheduler to distribute "slack" time intelligently through the schedule. Tasks durations were described as a range that specified the time for an agent to generate a solution of acceptable quality. Any time "remaining" (before the deadline) was then distributed through the schedule. For example, tasks that were yet to be decomposed, tasks that provided a key process product, and tasks that were currently assigned to a single agent but would need to be assigned to multiple sub-agents should the agent become unavailable were targeted for additional time allocation. The addition of slack time allowed the schedule to slip without the need to re-task agents and was particularly useful in the AIM domain where dependency on external events such as weather meant tasks would often run late. The addition of slack time allowed a number of different user policies to be explored in which the user was allowed to target where the slack was posted. This allowed the user to identify the trade-offs between different strategies (e.g., only allow slack to be posted on actions to be decomposed) and their impact on the overall schedule produced. These ideas will be explored further in the CoABS project MIATA demonstration.

Task Models

The main modification to the SWO algorithm was to allow it to reason with the PRFER and PAER task models described earlier. The tasks models allowed new tasks to be added and deleted from the priority queue as needed (e.g., if a bombing mission was assigned to an aircraft with insufficient range to reach the target, then an air-to-air refueling (AAR) task was added to the priority queue). The algorithm needed to keep track of these dependencies so that the AAR mission could be removed if the mission was assigned to a longer-range aircraft later. In addition, the dependencies between the different TSBs needed to be maintained and suggestions made as to ways to alter task durations should changes occur in the schedule (e.g., alter the configuration of the food drop should the duration of the mission be reduced). Several global measures were also added to allow the user to provide guidance in the scheduling process. These included estimates of resource breakdown (e.g., use no more than 30% of assets against the IAD) and durations (e.g., phase 1 should be run at least 200 hrs before moving to phase 2). These additions allowed DEOS to generate very good schedules that took into account the user guidance without any corresponding slowdown in performance. In addition to solution speed, the task models provided a key link between the generation and execution steps. For example, if an activity failed during execution, it was immediately obvious which tasks were affected and which could be re-tasked with minimum knock-on effects (e.g., there is no point in re-tasking a mission if the time to a newly located SAM battery is after the SAM battery has moved off). Subject Matter Experts (SMEs) in both the AIM and EBO programs claimed this was a major advantage over current systems.

Schedule Packing

Several experiments were conducted with the schedule pack algorithm [6, 5, 11] to identify ways it could be improved to provide better solutions.

- The non-systematic nature of schedule packing allows it to quickly find good areas of the search space. However, the ability to quickly move about the search space is a double-edged sword and schedule pack can sometimes quickly diverge from good solutions it finds, rather than refining them. Intensification (a term borrowed from Tabu search [22]) allows schedule packing to remember previous good solutions and to periodically restart the search from these. The effect of intensification is to focus the search more tightly on areas of the search space that have already produced good solutions. The results were very dramatic, allowing the scheduler to find solutions never seen before. On a number of benchmark production scheduling problems, this approach is believed to have produced the optimal solution. In addition the run times to identify high-quality solutions have been reduced from many hours to several minutes.

- In many domains there is a need to develop cyclic schedules in which new instances of a task are introduced periodically (e.g., wing assembly at Boeing). The schedule packing algorithm was successfully extended to allow it to handle these types of schedules with the additional benefit that the speed of the algorithm allowed execution changes and perturbations in the schedules to be handled very quickly. This is particularly important in organizations such as Boeing where the solutions to unexpected problems (i.e., updated schedules) should be available within the shift to avoid idling a large number of people [13], [27].

Heuristic Discrepancy Search

In solving a very large problems, heuristics or rules of thumb are often used. Heuristics guide the search toward regions of the space that are likely to contain solutions. Heuristic search is subject to the "early mistakes" problem where an early bad choice can cause expensive effort to be wasted in barren regions of the search space. Traditional backtracking strategies, like Depth First Search, usually do not recover from such early mistakes in time. If the number of "wrong turns" is small, they can be corrected by systematically searching paths that differ from the heuristic path at most a small number of decision points (discrepancies from the heuristic). Limited Discrepancy Search [23] searches the space in order of increasing discrepancy count, i.e., it searches the heuristic path first, then all paths that disagree with the heuristic in at most one decision, then two, etc. until it finally explores the whole search space. LDS undoes the early mistakes, often allowing better solutions to be found.

We extended the basic LDS algorithm to develop a new algorithm that allows for weighted values to be added to the arcs of the search tree. The new algorithm is referred to as Weighted Discrepancy Search (WDS) [1]. In WDS the heuristic preference is expressed as a fractional weight instead of just a 0-1 decision. The space is searched in order of decreasing weight instead of increasing number of discrepancies. Only nodes with weight greater than or equal to some cutoff value are explored in each iteration. We have derived a method to obtain optimal policies for progressively relaxing the cut-off for WDS to extend the search space as time allows.

Experimental results showed that, in comparison with LDS on randomly generated trees with branching factors of 2 and 3, WDS finds solutions while exploring 30% fewer modes than LDS on trees with a branching factor of 2 and 50% fewer on trees with a branching factor of 3. We also identified a number of dependencies amongst parameters in the search space and were able to reduce the number of parameters needed to control the search algorithm from (depth, branching factor, heuristic probability, mistake probability) to (depth, branching factor, heuristic function) .

Tractable Reasoning

We attempted to semantically characterize a large class of inferences that can be done quickly. The goal was to use this information to develop "anytime" approximate reasoning systems that are guaranteed to perform a broad class of inferences quickly, and which then do more complicated reasoning as time allows. The results of the effort were mainly theoretical but progress was made in the following areas:

- We developed a new nondeterministic type of semantic characterization of approximate reasoning systems and proved soundness and completeness results for our tractable reasoning approach [7]. To our knowledge, this type of semantics has never been proposed before, and it appears to hold great promise in understanding tractable reasoning systems.
- We proved several theorems that showed that the semantics supports a kind of iterative refinement of models. A system can start with a partial model and gradually make it more concrete as the reasoning process continues. We proved that the process will never preclude possibilities that actually remain open in so doing. The effort showed that reasoning systems characterized by these semantics are tractable, and showed how to incrementally extend them to more complete reasoners while never sacrificing polytime behaviors. In addition we proved a number of results relating our approach to others that have been put forward, showing that ours handles a larger class of reasoning problems without giving up tractability.
- We extended our framework from propositional logic to full first-order logic. This involved the synthesis of ideas developed for the propositional case with our ideas about context-limited reasoning [15]. The result is a framework that supports bounded, tractable reasoning within a limited context, but also provides semantic markers indicating where reasoning has been cut off. These markers allow the system to continue where it left off, should more resources become available for pursuing a line of reasoning. A paper describing those results is in preparation.

Phase Transitions

One aspect of the process management problem of particular interest to the project concerns how teams of agents should be formed and tasked. In particular, we observed that the phenomenon of phase transitions – where a problem moves from being easy to difficult to solve – provides possible guidance for this process. If a task is divided in the wrong way it can cause the whole problem to become dramatically harder to solve than it might otherwise be.

We showed that, in satisfiability problems, the number of unary prime implicates (UPIs) increases dramatically approaching the phase transition (from the satisfiable side). More generally, sharp increases in short implied constraints between a small number of variables (no-goods) occur approaching phase transitions. Turning this observation around suggests using this phenomenon to monitor the progress of time-bounded optimizing solvers. The idea is to use the emergence of large numbers of short no-goods to detect when the solver is about to reach a computational cliff [31, 20].

In an environment with multiple computational units, such information is useful because it allows better decisions to be made as to how to distribute the workload and computational resources between the units. If a unit is projected to reach a computational cliff, a better informed decision can be made about whether to allocate more resources or decide that further progress is unlikely and instead reallocate the resources in question. In addition, we showed that the structure of the no-goods that emerge during the reasoning process can contain deep information about the structure of the problem and this may be used to improve the computational organization [21] – for example, to changing the way the units are formed into collaborative coalitions.

5 Empirical Studies

This section describes the empirical studies undertaken during this program. As stated earlier, we made contacts with two other projects (EBO and AIM) that provided frameworks through which we evaluated our ideas. These domains were interesting because, in addition to providing real-world operational problems to "keep up honest", they contained several features that stressed the technology in interesting ways.

The EBO domain uses a complicated multi-objective function, trying to generate schedules with a low expected attrition rate, high probability of damage, and short makespan. These objectives are, to a certain extent, in conflict. For example, the easiest way to reduce the makespan is to send aircraft out on the next available mission as soon as it returns. However, this means aircraft may be attacking targets for which they have unacceptably high attrition rates (e.g., A-10s against SAM sites) or that the weapons the aircraft can carry have a very low probability of damage (e.g., a MK-84 gravity bomb against a hardened bunker). Alternatively, if the best (i.e., lowest attrition rate and highest probability of damage) aircraft are always selected then every mission would be flown at night by F-117s, resulting in unacceptably long makespans. In addition to the obvious constraints on missions there were constraints on the overall mix of missions. For example, constraints of the form "no more than 40% of the assets should be tasked against the IAD" force the scheduler to select the "best" missions out of the entire target set. The strike missions are supported by other assets (e.g., AAR, SEAD) which also need to be factored in. An aircraft flying a SEAD mission is not available to attack targets, which may impact on the makespan of the schedule.

Targets can be missed or damaged less than required and thus need to be added back into the target list. Planned missions may thus need to be retasked to deal with the missed target. These retasked targets need to be added into the schedule while causing the minimum amount of disruption. Each of these issues pose significant technical challenges and were successfully addressed.

Schedules generated in the AIM domain need to have the flexibility to deal with large numbers of unexpected situations. The unexpected events include changes in both the mission and the environment. The scheduler needed to make decisions such as, "should an agent be restarted with updated information or allow it complete its task and deliver a lower quality product". For example, if an agent is developing tomorrow's ISR asset list and all U2 aircraft are grounded, the list it produces is still valid for other resource types. If other agents are dealing with F-4 mission planning, they can use the product produced and should not be delayed if possible. Balancing the quality of information against the costs involved in generating it were seen as key to the ISR planning problem. The temporal robustness built into the schedules allowed the scheduler to move task allocations around, insert new tasks and modify existing task structures as new events needed to be dealt with.

5.1 The EBO Domain

The main test domain for the DEOS system has been the EBO domain. EBO provides a good testbed due to the large number of aircraft/weapon combinations (700 missions, 200 aircraft of 15 different types, and 120 different weapon systems). As describes above the optimization function is a multi-objective one and the best schedule identified by DEOS completes all 700 missions in 47 hours with a high probability of destruction and an expected loss rate of less than 1%. To date the DEOS results are the best (interms of attrition and makespan) for these problems and are comparable to, if not better than, those developed by current USAF mission planners.

One advantage of the SWO approach is that it assigns a blame score for tasks that are handled badly. In the case of mission management, this could be due to a lateness, low probability of damage, or high expected attrition rate. It is often the case that to obtain a good overall schedule with the available resources some tasks need to be handled badly (i.e., they need to be sacrificed). It is possible to handle the sacrificed tasks better but only at the expense of making the overall schedule worse. The blame score can indicate to the human planners where additional resources are needed to solve all the tasks more efficiently.

5.2 The AIM Domain

The AIM domain provided the main test for the PAER task and information flow models. This was a very good domain as it is dominated by the large amounts of information that need to be collected, analyzed, disseminated and updated. While the domain contains fewer agents than ACP, there is a far higher level of dynamic change. For example, if a collection task is looking for T-72 tanks at a given location and identifies a SAM missile launcher, this needs to be communicated to the appropriate bodies, collection and analysis assets re-prioritized, and tasks re-assigned. The PAER models were able to capture the different events occurring in the domain at their different levels of abstraction and show the dependencies and links between the tasks being performed. While the DEOS system is a work in progress, an initial version was successfully demonstrated in August 1999 to the ISR community. The initial demonstration tackled problems varying from lost assets to large scale re-plans, resource updates, and changes in mission objectives. The feedback from the SMEs was that DEOS allows processes to be identified, resourced and updated far faster than current systems. In particular DEOS allowed the commanders to identify the trade-offs between re-tasking agents within the same process and generating new process instances. The information model was also identified as a key element, as it allowed commanders to identify the trade-offs in using process products of lower quality in exchange for having the results earlier (e.g., knowing the ISR plan calls for either satellites or U2's provided the track planners with information that could kick-start their process before the final decision as to which assets should be used). The ideas initially prototyped in the AIM domain are now being applied to and extended for the process-management issues in DARPA's CoABS project. A demonstration of these ideas is planned for early 2001 within a disaster-relief scenario.

6 External Interactions

This section describes the interactions the project has undertaken with other DARPA projects, groups and commercial organizations. We developed contacts with several DARPA and AFRL projects that could provide example command and control hierarchies. The two most important were DARPA's AIM program and AFRL's EBO program.

DARPA AIM Program

The AIM program provided various hierarchies concerned with the integration of collection, processing, exploitation and dissemination activities involved in ISR management. These hierarchies are characterized by being highly dynamic and reactive to changes occurring in the clients' collection needs and the availability of assets. The hierarchies also contain many agents with large variations in their capabilities. As described earlier, we applied the PAER task model and scheduling robustness ideas to the DEOS scheduling system that formed one of the core elements of the SWIM system jointly developed with SRI. The DEOS system was demonstrated to several ISR SMEs in August 1999 as part of AIM's process-management thread. The demonstration showed DEOS generating schedules for in excess of 50 software agents and rescheduling due to external events. The external events ranged from simple failure of assets to collect the required data through to large scale reschedules caused by changes to commander's guidance, information collection reprioritization, and changes to major deadlines (e.g., ground-war start). The demonstration showed that DEOS was able to significantly reduce rescheduling time and recover from events that the SMEs [4] claimed current systems could not handle.

AFRL EBO Program

The EBO program provided various hierarchies concerned with mission planning (i.e., the selection of weapon and aircraft pairings for a specified target list). These hierarchies are characterized by the need to coordinate and synchronize large numbers of assets belonging to different groups and commands (e.g., tanker aircraft, SEAD, airlift aircraft, AWACS). As described earlier, we applied the PRFER task model and scheduling robustness ideas to the DEOS system that was developed for the EBO jump-start demonstration. The DEOS system was demonstrated at Langley AFB in November 2000. The demonstration showed DEOS generating schedules for several different target sets, taking into account constraints including priority, weight of effort, sequencing and time windows. The demonstration showed that DEOS is able to generate good schedules[5] very quickly and can react quickly to changes in resource levels. DEOS' ability to provide feedback to the human planners on the impact of changes and to provide "what if" support was seen as vital to future mission-management tools.

[4] Cmdr Carol Thompson, Mike Kramer

[5] Per comments from Col. Buster McCrabb, the EBO SME.

Commercial Involvement

In addition to the involvement with other research projects, we devoted time and effort to making links with groups in industry. These included the mission-management groups at Lockheed-Martin in Mannassas and at Boeing in Philadelphia. DEOS was selected as the preferred technology for the mission-management software for the CVN-77 "Nimitz-after-next" aircraft carrier. It is expected that work will begin in the first quarter of 2001. Boeing is currently evaluating the DEOS system for inclusion in the MOM mission-management framework and for inclusion in an EBO-based effort they are conducting for the Navy.

7 Summary and Future Work

The project has shown that explicit modeling of information, its content and the ways in which it changes can provide a powerful way of handling large distributed problems. In addition, the use of generic task description templates (PRFER and PAER task models) greatly improves the agent-tasking process by making explicit the constraints and dependencies between tasks. The PRFER and PAER task models themselves allow scheduling algorithms to understand the trade-offs that are possible within a task and to identify ways in which tasks can be modified to suit the changing environment. For example, if a mission to drop food pallets is delayed, then reconfiguring the mission to an airdrop rather than a landing may recover the lost time. The task description language itself was developed to allow it to describe both the tasks and the capabilities of the agents that could be assigned to process the task. This allows for a far more efficient and less cumbersome match-making process in assigning agents to tasks.

Research in the information-processing aspects of process management highlighted several new potential areas of research interest. Of particular interest is the problem of phase transitions. These are "computational cliffs" that occur naturally in problems and represent the point at which problems transition from being fairly easy to solve to be being very difficult to solve. Unfortunately, most of the problems people look at tend to fall in this transition region and hence the potential benefits of finding computational improvements are very high. One area in which this problem can be acute is the development of teams to solve large distributed problems. Dividing the problems arbitrarily can lead to very inefficient solutions that take too long times to produce. We are exploring the problem of phase transitions in the context of autonomous team building through a project funded by DARPA's ANTs program, which is directly founded on ideas developed on this project.

The technologies and ideas developed during the project have been successfully applied to problems in the areas of mission planning and ISR task collection and processing. The DEOS system proved to offer faster and more flexible solutions than were available using current technologies. Future research for the DEOS system itself includes adding capabilities to deal with adversarial situations in which agents are attempting to interfere with the aims of the process. This would be particularly useful in situations in which the U.S. forces do not have overwhelming superiority and need to have dynamic and responsive plans capable of dealing with a wide variety of enemy options.

The research advances reported here have allowed us to make progress in the three key areas: information content, flow and processing. These ideas are now being exploited through several technology-transfer paths. In particular, the DEOS system has been selected by Lockheed-Martin for the mission management component of the CVN-77 "Nimitz-carrier-after-next" project. Work is due to begin in early 2001. Other initiatives with Boeing and Bath Iron Works offer further potential applications of this technology. Our aim is also to find further applications [19] through close cooperation with On Time Systems, a commercial company spun out of the University of Oregon.

References

[1] Bedrax-Weiss, T.: *Optimal Search Protocols*, PhD thesis, University of Oregon, Eugene, OR, 1999.

[2] Berry, P.M. and Drabble, B.: "SWIM: An AI-based System for Workflow Enabled Reactive Control", in *Proceedings of the Workshop on Workflow and Process Management* held as part of the International Joint Conference on Artificial Intelligence (IJCAI-99), (eds B. Drabble and M. Ibrahim), IJCAI Inc, August, 1999.

[3] Berry P.M. and Drabble, B.: *Intelligent Workflow for Collection Management*, AI Center, SRI International, Technical Report 1710, Menlo Park, CA, 1998.

[4] Berry, P.M. and Drabble, B.: "SWIM: An AI-based System for Organization Management", in *Proceedings of the Second International NASA Workshop on Planning and Scheduling for Space*, San Francisco, CA, 16th-18th March 2000.

[5] Clements, D., Crawford, J., Joslin, D., Nemhauser, G., Puttlitz, M. and Savelsbergh, M.: "Heuristic Optimization: A Hybrid AI/OR Approach", in *Proceedings of the workshop on Industrial Constraint-Directed Scheduling*, 1997. (Held in conjunction with CP'97, Schloss Hagenberg, Austria.)

[6] Crawford, J.: "An Approach to Resource Constrained Project Scheduling", *Proceedings of the Artificial Intelligence and Manufacturing Research Planning Workshop*, June 1996, AAAI Press, SIGMAN.

[7] Crawford, J., and Etherington, D.W.: "A Non-Deterministic Semantics for Tractable Reasoning", in *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, Madison, WI, July, 1998.

[8] Drabble, B.: "Task Decomposition Support to Reactive Scheduling", *Fifth European Conference on Planning (ECP-99)*, Springer Verlag Press, New York, NY, September, 1999.

[9] Drabble, B. and Clements, C.: "Make-span Scheduling for Assembly Tasks: Extended Abstract", in the working papers of the *NASA Workshop on Planning and Scheduling for Space*, October 1997, Oxnard, CA.

[10] Drabble, B. and Lydiard, T.J. and Tate, A.: "Workflow Support in the the Air Campaign Planning Process", in *Proceedings of the Workshop on Interactive and Collaborative Planning*, held as part of Fourth International Conference on Artificial Intelligence Planning Systems (AIPS-98), (eds. K. Myers and G. Ferguson), CMU, Pittsburgh, PA, June, 1998.

[11] Drabble, B., McVey, C.B. and Clements, D.P.: "Agile Aircraft Manufacturing and Assembly", in *Proceedings of the Symposium on "Planning, Scheduling, and Control for Aerospace*, at the World Congress on Automation (WAC-2000) Wailea, Maui, June 2000.

[12] Drabble, B., Tate, A. and Dalton, J.: "Repairing Plans On-the-fly", in the working papers of the *NASA Workshop on Planning and Scheduling for Space*, October 1997, Oxnard, CA.

[13] Draper, D., Jonsson, A., Clements, D. and Joslin, D.: "Cyclic Scheduling", in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, Stockholm, Sweden, July, 1999.

[14] Etherington, D.W.: "What Soes KR Have to Say to AI", extended abstract for an invited talk, *Fourteenth National Conference on Artificial Intelligence*, Providence, RI, July, 1997.

[15] Etherington, D.W. and Crawford, J.M.: "Toward Efficient Default Reasoning", in *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, Portland, OR, August 1996.

[16] Ferguson, G., Allen, J. and Miller, B.: "TRAINS-95: Towards a Mixed Initiative Planning Assistant", in *proceedings of the Third International Conference on Planning (*AIPS-96*)*, Edinburgh, May 1996, (ed. Drabble, B.), AAAI Press, Menlo Park, CA.

[17] Georgeff, M.P. and Ingrand F.F.: "Decision-Making in an Embedded Reasoning System", in *Proceedings of the Eleventh International Joint Conference on AI*, Detroit, MI, 1989.

[18] Ginsberg, M.: "GIB: Steps Towards and Expert-Level Bridge-Playing Program", in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, Stockholm, Sweden, July, 1999.

[19] Ginsberg, M.L., Etherington, D.W, and Drabble, B,: "Can Search Play a Role in Practical Applications", in *Proceedings of AI Meets the Real World '98*, University of Connecticut, Stamford, CT. September, 1998.

[20] Ginsberg, M.L. and Parkes, A.J.: "Satisfiability Algorithms and Finite Quantification", in *Proceedings of Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000)*, Breckenridge, Colorado, USA, 12-15 April 2000.

[21] Ginsberg, M.L., Parkes, A.J. and Roy, A.: "Supermodels and Robustness", in *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, Madison, WI, AAAI Press, July, 1998.

[22] Glover, F. and Laguna Manual.: *Tabu Search*, Kluwer Academic Publishers, Boston, 1997, ISBN: 0-7923-9965-X.

[23] Harvey, W.D. and Ginsberg, M.L.: "Limited Discrepancy Search", in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95*, Montreal, Canada, Morgan Kaufmann Inc, August 1995.

[24] Hess, D.: *Theater Battle Management Command, Control, Communications, Computer and Intelligence Architecture (TBM C4I* Air Operations Center (AOC), Report prepared by Government personnel and Mitre Corporation under Contract for Project 6970.

[25] Jarvis, P., Lydiard T.J. and Drabble B.: "Realizing Real Commercial Benefit from Workflow: A Report from the Trenches", in *Proceedings of the Workshop on Agent Based Systems in Business* held as part if the Sixteenth National Conference on Artificial Intelligence (AAAI-99), Orlando, FL, July 1999.

[26] Joslin, D.E. and Clements, D.P.: "Squeakywheel Optimization", in *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, Madison, WI, AAAI Press, 1998.

[27] Joslin, D.E. and Roy, A.: "Exploiting Symmetry in Lifted CSPs", in *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, Providence, RI, July, 1997.

[28] Myers, K.L.: *The ACT Editor User's Guide*, AI Center, SRI International, Menlo Park, CA, 1993.

[29] McVey, C.B., Clements, D.P., Massey, B.C., Parkes, A.J. and Drabble, B.: "Worldwide Aeronautical Route Planner: Optimal fuel usage flight Planning" in *Proceedings of the Symposium on "Planning, Scheduling, and Control for Aerospace* at the World Congress on Automation (WAC-2000) Wailea, Maui, June 2000.

[30] Myers, K.L.: "A Procedural Knowledge Approach to Task-Level Control", in *Proceedings of the Third International Conference on AI Planning Systems (AIPS-96)*, Edinburgh, May 1996, (ed. Drabble, B.), AAAI Press, Menlo Park, CA.

[31] Parkes, A.J.: "Clustering in the Phase Transition", in *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, Providence, RI, July, 1997.

[32] Smith, S.F.: "OPIS: A Methodology and Architecture for Reactive Planning", in *Intelligent Scheduling*, (eds. Zweben, M. and Fox, M.), 1994, Morgan Kaufmann Publishers Inc, San Francisco, CA, USA.

[33] Swartout, W.R. and Gil, Y.: "EXPECT: A User-Centered Environment for the Development and Adaptation of Knowledge-Based Planning Aid", in *Advanced Planning Technology: Technology Achievements of the* ARPA*/Rome Laboratory Planning Initiative*, (ed. Tate, A.), 1996, AAAI Press, Menlo Park, CA.

[34] Swartout, W.R., Patil, R., Knight, K., and Russ, T.: "Toward Distributed Use of Large-Scale Ontologies", in *Proceedings of the 1996 Banff Workshop on Knowledge Acquisition.*

[35] Tate, A., Drabble, B. and Kirby, R.: "O-Plan2: An Open Planning Architecture for Command, Planning and Control", in *Intelligent Scheduling*, (eds. Zweben, M. and Fox, M.), 1994, Morgan Kaufmann Publishers Inc, San Francisco, CA.

[36] Valente, A., Swartout, W. and Gil, Y.: *A Representation and Library for Objectives in Air Campaign Plans.* Technical Report, USC–Information Sciences Institute, 1996.

[37] Wilkins, D.E. and Desimone, R.V.: "Applying an AI Planner to Military Operations Planning", in *Intelligent Scheduling*, (eds. Zweben, M. and Fox, M.), 1994, Morgan Kaufmann Publishers Inc, San Francisco, CA.

8 Published Papers

This section describes the papers published during the project.

1. Bedrax-Weiss, T.: *Optimal Search Protocols*, PhD thesis, University of Oregon, Eugene, OR, 1999.

   Abstract:
   This thesis provides an overview of the Limited Discrepancy Search (LDS) that was discovered at CIRL and of Weighted Discrepancy Search (WDS) which was developed to overcome some of the shortcomings of LDS. LDS has been generalized to the case where the heuristic preference is a fractional weight instead of just a 0-1 decision. In weighted discrepancy search, the space is searched in order of decreasing weight instead of increasing number of discrepancies. Only nodes with weight greater than or equal to the cutoff are explored in each iteration. Thie thesis describes the method to obtain optimal cutoff policies for WDS.

2. Berry, P.M. and Drabble, B.: "SWIM: An AI-based System for Workflow Enabled Reactive Control", in *Proceedings of the Workshop on Workflow and Process Management* held as part of the International Joint Conference on Artificial Intelligence (IJCAI-99), (eds B. Drabble and M. Ibrahim), IJCAI Inc, August, 1999.

   Abstract:
   This paper describes the initial development of the Smart Workflow for ISR Management (SWIM) system, which was designed to enable complex systems, businesses and software, to be controlled within a workflow management paradigm. SWIM extends the workflow paradigm to respond to dynamic and uncertain environments by viewing the control processes themselves as being dynamic, evolving entities. The paper also describes several initial experiments that have been conducted in the ISR domain.

3. Berry P.M. and Drabble, B.: *Intelligent Workflow for Collection Management*, AI Center, SRI International, Technical Report 1710, Menlo Park, CA, 1998.

   Abstract:
   This report describes the generic modeling methodology that was developed to design and represent the process of Advanced Information Surveillance, and Reconnaissance (ISR) Management (AIM). The AIM process consists of several activities and sequences of activities that must be performed to successfully build and maintain the ISR plan. The methodology is also intended to gather knowledge and expertise in the creation of process models, with the intention being to lead to the dynamic creation and adaption of processes to suit ever-changing environmental and mission conditions. The paper describes the development of the initial process models and how these were refined through interactions with ISR experts.

4. Berry, P.M. and Drabble, B.: "SWIM: An AI-based System for Organization Management", in *Proceedings of the Second International NASA Workshop on Planning and Scheduling for Space*, San Francisco, CA, 16th-18th March 2000.

   Abstract:
   This paper further reports on the development of the SWIM system described above. SWIM was applied to the domain of Information Surveillance and Reconnaissance (ISR), a highly reactive domain where continual and complex requirements for information acquisition, analysis and distribution must be satisfied within a temporal and resource constrained setting. The SWIM system comprises two main components: a process manager and the Dynamic Execution Order Scheduling system. Details of both of these components are provided in the paper.

5. Clements, D., Crawford, J., Joslin, D., Nemhauser, G., Puttlitz, M. and Savelsbergh, M.: "Heuristic Optimization: A Hybrid AI/OR Approach", in *Proceedings of the workshop on Industrial Constraint-Directed Scheduling*, 1997. (Held in conjunction with CP'97, Schloss Hagenberg, Austria.)

   Abstract:
   This paper describes a hybrid architecture, H-OPT, that combines Integer Programming (IP) for global optimization, and heuristic search techniques. The approach captures the most desirable features of each. A heuristic local search algorithm generates a large number of good feasible solutions quickly, and the IP solver is then used to combine the elements from those solutions into a better solution than the local search approach was able to find. Preliminary experimental results are very encouraging. The techniques described are very general, and should be applicable to a wide range of problems. The paper reports very promising results on a scheduling problem that arises in fiber-optic cable manufacturing. The heuristic approach can generate good solutions very quickly by itself, but in combination with the global IP optimization significant further improvement is possible. The hybrid approach also produces better quality solutions than a tabu search algorithm, and runs faster as well.

6. Crawford, J., and Etherington, D.W.: "A Non-Deterministic Semantics for Tractable Reasoning", in *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, Madison, WI, July, 1998.

   Abstract:
   Unit resolution is arguably the most useful known algorithm for tractable reasoning in propositional logic. Intuitively, if one knows a, b, and a and b implies c, then c should be an obvious implication. However, devising a tractable semantics that allows unit resolution has proven to be an elusive goal. This paper describes a 3-valued semantics for a tractable fragment of propositional logic that is inherently non-deterministic: the denotation of a formula is not uniquely determined by the denotation of the variables it contains. The paper shows that this semantics yields a tractable, sound and complete, decision procedure. The paper also describes a generalization of these semantics to a family of semantics, tied to Dalal's notion of intricacy, of increasing deductive power and computational complexity.

7. Drabble, B.: "Task Decomposition Support to Reactive Scheduling", *Fifth European Conference on Planning (ECP-99)*, Springer Verlag Press, New York, NY, September, 1999.

   Abstract:
   This paper describes the development of an intelligent tasking model which has been designed to enable complex systems, human agents, and software

agents, to be tasked and controlled within a reactive workflow management paradigm. The Dynamic Execution Order Scheduler (DEOS) extends the current workflow paradigm to allow tasking in dynamic and uncertain environments by viewing the planning and scheduling tasks as being integrated and evolving entities. The paper describes the application of the DEOS to the domains of Air Campaign Planning (ACP) and ISR management. These are highly reactive domains in which new tasks and priorities are identified continuously and plans and schedules are generated and updated within a time and resource-constrained setting.

8. Drabble, B. and Clements, D.: "Makespan Scheduling for Assembly Tasks: Extended Abstract", in the working papers of the *NASA Workshop on Planning and Scheduling for Space*, October 1997, Oxnard, CA.

   Abstract:
   This paper describes several scheduling techniques that have been applied to areas including aircraft assembly and fiber-optic cable manufacturing. The techniques described are Limited Discrepancy Search (LDS) and the Doubleback Optimization. These techniques are acting as a test bed for further developments including a "probabilistic" version of LDS and the ability to handle more complex concepts such as release times and delivery dates. The paper provides details of the techniques and describes ways in which they could be used for applications such as assembly, integration and verification.

9. Drabble, B. and Lydiard, T. and Tate, A.: "Workflow Support in the the Air Campaign Planning Process", in *Proceedings of the Workshop on Interactive and Collaborative Planning*, held as part of Fourth International Conference on Artificial Intelligence Planning Systems (AIPS-98), (eds. K. Myers and G. Ferguson), CMU, Pittsburgh, PA, June, 1998.

   Abstract:
   This paper describes a model developed to describe the Air Campaign Planning (ACP) process and to characterize the capabilities of systems and technologies being developed to support this process within the DARPA/Rome Laboratory Planning Initiative (ARPI). Verb/noun(s)/qualifier(s) statements are used to define a process ontology that details the activities that take place in generating ACP plans (e.g., refine, issue, analyze), a set of process products flowing through the process, and the capabilities of agents and systems. The paper describes the motivation behind developing the verb/noun(s)/qualifier(s) model, defines the different entities in the model, and shows how the model can be used to define the capabilities and process plans steps of the ACP process and in particular the USAF's Air Campaign Planning Tool (ACPT).

10. Drabble, B., McVey, C.B. and Clements, D.P.: "Agile Aircraft Manufacturing and Assembly", in *Proceedings of the Symposium on "Planning, Scheduling, and Control for Aerospace*, at the World Congress on Automation (WAC-2000) Wailea, Maui, June 2000.

Abstract:
This papers describes several intelligent scheduling techniques that have been applied to problems in manufacturing and assembly. The techniques have been evaluated in domains including aircraft wing manufacturing, fiber-optic cable manufacturing, submarine construction and CD manufacturing. The paper describes two particular scheduling techniques: schedule packing and squeaky wheel optimization. The techniques have resulted in a number of major improvements including reduction in make-span of up to 50%, improvement of throughput by 40%, reduction in costs of 20%, and the ability to tackle problems up to 20 times larger. The paper provides an overview of these techniques and describes a number of case studies from Boeing, Electric Boat, and Lucent Technologies. The paper concludes with a description of the impact these techniques have had in each of these organizations and provides pointers to other allied scheduling problems in the military domain.

11. Drabble, B., Dalton, J, and Tate, A.: "Repairing Plans On-the-fly", in the working papers of the *NASA Workshop on Planning and Scheduling for Space*, October 1997, Oxnard, CA.

Abstract:
Several methods for repairing plans to account for execution failures and changes in the execution situation are described. They were first developed for the Optimum-AIV planner designed to support spacecraft assembly, integration and verification at European Space Agency, and later deployed for Ariane IV payload bay AIV. This system was itself based on the Nonlin and O-Plan planning algorithms and plan representation. This paper describes the algorithms used for plan repair in O-Plan and gives an example of their use.

12. Draper, D., Jonsson, A., Clements, D. and Joslin, D.: "Cyclic Scheduling", in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, Stockholm, Sweden, July, 1999.

Abstract:
This paper describes the problem of cyclic schedules such as arise in manufacturing. It introduces a new formulation of this problem that is a very simple modification of a standard job shop scheduling formulation which enables the use of existing constraint reasoning techniques to generate cyclic schedules. The paper presents evidence for the effectiveness of this formulation, and describe ex-

tensions for handling multiple-capacity resources and for recovering from breaks in cyclic schedules.

13. Etherington, D.W.: "What Does Knowledge Representation Have to Say to Artificial Intelligence", extended abstract for an invited talk, *Fourteenth National Conference on Artificial Intelligence*, Providence, RI, July, 1997.

    Abstract:
    In recent years, the subarea of Knowledge Representation and Reasoning (KR) has become more and more of a discipline unto itself, focusing on artificial problems while other areas of AI have tended to develop their own representations and algorithms. There are signs that this is changing, however. This explored what the current state of KR has to offer to AI.

14. Ginsberg, M.: "GIB: Steps Towards an Expert-Level Bridge-Playing Program", in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, Stockholm, Sweden, July, 1999.

    Abstract:
    This paper describes GIB, the first bridge-playing program to approach the level of a human expert. GIB finished twelfth in a hand-picked field of thirty-four championship level players at an invitational event at the 1998 World Bridge Championships. The paper provides a basic overview of the algorithms used, describe their strengths and weaknesses, and present the results of experiments comparing GIB to both human opponents and other programs.

15. Ginsberg, M., Drabble, B., and Etherington, D.W.: "Can Search Play a Role in Practical Applications", in *Proceedings of AI Meets the Real World '98*, University of Connecticut, Stamford, CT. September, 1998.

    Abstract:
    Received wisdom is that search in ineffective in fielded AI systems; it is argued realistic problems are simply too large to allow effective search. This paper suggests that this is because search-spaced techniques fall prey to early mistakes, making fatal errors sufficiently early in the search that back-tracking techniques are incapable of correcting them. The paper describes four general techniques that have been used in practical applications to circumvent this difficulty: limited discrepancy search, schedule packing, squeaky wheel optimization and relevance bounded learning. The paper explains how each technique avoids the early mistake problem, and describes the impact of the techniques on specific problems of practical interest.

16. Ginsberg, M.L. and Parkes, A.J.: "Satisfiability Algorithms and Finite Quantification", in *Proceedings of Seventh International Conference on Principles of*

Knowledge Representation and Reasoning (KR2000), Breckenridge, Colorado, USA, 12-15 April 2000.

Abstract:
Three observations are made with regard to the application of algorithms such as WSAT and RelSAT to problems of practical interest. First, a specific calculation ("subsearch") is identified that is performed at each inference step by any existing satisfiability algorithm. It is then shown that for realistic problems, the time spent on subsearch can be expected to dominate the computational cost of the algorithm. Finally, a specific modification to the representation that exploits the structure of naturally occurring problems and leads to exponential reductions in the time needed for subsearch is presented.

17. Ginsberg, M.L., Parkes, A.J. and Roy, A.: "Supermodels and Robustness", in *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, Madison, WI, AAAI Press, July, 1998.

    Abstract:
    When search techniques are used to solve a practical problem, the solution produced is often brittle, in the sense that small execution difficulties can have arbitrarily large effects on the viability of the solution. The AI community has responded to this difficulty by investigating the development of "robust problem solvers" that are intended to be proof against this difficulty. This paper argues that robustness is best cast not as a property of the problem solver, but as a property of the solution. It introduces a new class of models for a logical theory, called *supermodels*, to capture this idea. Supermodels guarantee that the model in question is robust, and allow one to quantify the degree to which it is so. The paper investigates the theoretical properties of supermodels, showing that finding supermodels is typically of the same theoretical complexity as finding any models. Experimentally, it shows that the supermodel problem exhibits phase transition behavior similar to that found in other satisfiability work.

18. Lydiard T., Jarvis, P.and Drabble B.: "Realizing Real Commercial Benefit from Workflow: A Report from the Trenches", in *Proceedings of the Workshop on Agent Based Systems in Business* held as part if the Sixteenth National Conference on Artificial Intelligence (AAAI-99), Orlando, FL, July 1999.

    Abstract:
    This paper presents a case study of a UK banking business and the problems it encountered in deploying a workflow system. The paper sets out the research issues raised by this case study and discusses how emerging AI technologies could be exploited in satisfying them. The AI technologies discussed include information-gathering planning, general planning, and scheduling. The paper concludes by encouraging the formation of partnerships between workflow users,

workflow vendors and AI researchers. Such partnerships will give researchers access to real problems that can be used to demonstrate the scalability of their work and provide evidence that will encourage vendors and users to exploit the technologies. The feedback will also guide researchers on where further research should be focused.

19. Joslin, D. and Clements, D.: "Squeaky Wheel Optimization", in *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, Madison, WI, AAAI Press, 1998.

    Abstract:
    This paper describes a general approach to optimization which is referred to as "Squeaky Wheel" Optimization (SWO). In SWO, a greedy algorithm is used to construct a solution which is then analyzed to find the trouble spots, i.e., those elements, that, if improved, are likely to improve the objective function score. That analysis is used to generate new priorities that determine the order in which the greedy algorithm constructs the next solution. This Construct/Analyze/Prioritize cycle continues until some limit is reached, or an acceptable solution is found.

20. Joslin, D. and Roy, A.: "Exploiting Symmetry in Lifted CSPs", in *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, Providence, RI, July, 1997.

    Abstract:
    When search problems have large-scale symmetric structure, detecting and exploiting that structure can greatly reduce the size of the search space. Previous work has shown how to find and exploit symmetries in propositional encodings of constraint satisfaction problems (CSPs). This paper describes problems that have more compact "lifted" (quantified) descriptions from which propositional encodings can be generated. The paper describes an algorithm for finding symmetries in lifted representations of CSPs, and shows sufficient conditions under which these symmetries can be mapped to symmetries in the propositional encoding. Using two domains (pigeonhole problems, and a CSP encoding of planning problems), the paper shows that experimentally that the approach of finding symmetries in lifted problem representations is a significant improvement over previous approaches that find symmetries in propositional encodings.

21. McVey, C., Clements, D., Massey, B., Parkes, A. and Drabble, B.: "Worldwide Aeronautical Route Planner: Optimal fuel usage flight Planning" in *Proceedings of the Symposium on "Planning, Scheduling, and Control for Aerospace* at the World Congress on Automation (WAC-2000) Wailea, Maui, June 2000.

Abstract:
This paper provides an overview of the WARP system, its interface and basic search methods. The paper focuses on the rapid determination of minimal fuel routes for aircraft flying from any source point to any destination point on the earth. The paper describes the different routing methods including point to point, airways and via navigational beacons. The fundamental problem is the same (though progressively more constrained) for aircraft, ship, or land vehicle routing, and the same basic algorithms apply in each case.

22. Parkes, A.J.: "Clustering in the Phase Transition", in *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, Providence, RI, July, 1997.

    Abstract:
    Many problem ensembles exhibit a phase transition that is associated with a large peak in the average cost of solving the problem instances. However, this peak is not necessarily due to a lack of solutions: indeed the average number of solutions is typically exponentially large even at the peak. This paper studies this issue for the satisfiability transition in Random 3SAT. It shows that a significant subclass of instances emerges as the phase transition is crossed. These instances are characterized by having about 85–95% of their variables occurring in unary prime implicates (UPIs), with their remaining variables being subject to few constraints. In such instances the models are not randomly distributed: they all lie in a cluster that is exponentially large, but still admits a simple description. Studying the effect of UPIs on the local-search algorithm WSAT showed that these "single-cluster" instances are harder to solve, and we relate their appearance at the phase transition to the peak in search cost. In particular, the paper relates the hardness to so-called "failed clusters": exponentially large regions of the search space that would be a solution cluster were it not for a few constraints that happen to remove all the solutions. Such failed clusters are very good at trapping local search and causing it to fail. The significance is that detecting such causes of failure offers the hope of finding remedies.

Glossary of Acronyms and Abbreviations

AAR: air to air refueling

ACM: air combat mission

ACO: air component order

ACP: air campaign planning

AFB: air force base

AFRL: Air Force Rome Laboratory

AIM: advanced ISR management

AMC: Air Mobility Command

ANTS: Autonomous Negotiating Teams

ARPI: DARPA/Rome Laboratory Planning Initiative

ATO:: air tasking order

AWACS: airborne warning and control system

CAP: combat air patrol

CAS: close air support

CoABS: Control of Agent Based Systems

CRUA: create, read, update and authorize

CVN: nuclear powered aircraft carrier

DARPA: Defense Advanced Research Project Agency

DD-21: 21st century land bombardment destroyer

DEOS: Dynamic Execution Order Scheduling

EBO: Effects Based Operations

IAD: integrated air defense system

ISR: intelligence, surveillance and reconnaissance

JFACC: joint forces air component commander

JIPT: joint integrated prioritized

JIPTL: joint integrated prioritized target list

JTCB: Joint Targeting Coordination Board

LDS: limited discrepancy search

MIATA: mixed initiative agent tasking architecture

MOM: mission operations manager

PAER: plan, acquire, execute, report task model

PRFER: lan, ready, fly, execute, recover task model

PRS: procedural reasoning system

SAIC: Science Associates International Corporation

SAM: surface to air missile

SEAD: suppression of enemy air defenses

SME: subject matter expert

SWIM: Smart Workflow for ISR Management

SWO: squeaky wheel optimization

TSB: task specification block

VNQ: verb/noun/qualifier

WDS: weighted discrepancy search